EECS 16B Designing Information Devices and Systems II
Spring 2021 Discussion Worksheet Discussion 7B

The relevant notes for this discussion are Note 7B and Note 8.

1. System identification by means of least squares

Working through this question will help you understand better how we can use experimental data taken from a (presumably) linear system to learn a discrete-time linear model for it using the least-squares techniques you learned in 16A. You will later do this in lab for your robot car.

As you were told in 16A, least-squares and its variants are not just the basic workhorses of machine learning in practice, they are play a conceptually central place in our understanding of machine learning well beyond least-squares.

Throughout this question, you should consider measurements to have been taken from one long trace through time.

(a) Consider the scalar discrete-time system

$$x[i+1] = ax[i] + bu[i] + w[i] \tag{1}$$

Where the scalar state at time i is $x[i]$, the input applied at time i is $u[i]$ and $w[i]$ represents some external disturbance that also participated at time i.

Assume that you have measurements for the states $x[i]$ from $i = 0$ to m and also measurements for the controls $u[i]$ from $i = 0$ to $m-1$.

Set up a least-squares problem that you can solve to get an estimate of the unknown system parameters a and b.

Answer: Our model is of the form

$$x[i+1] = ax[i] + bu[i] + w[i]$$

where $w[i]$ is our error term and we are interested in a and b.

We have $[1, m]$ measurements, and so our least squares formulation is:

$$\begin{bmatrix} x[1] \\ x[2] \\ \vdots \\ x[m] \end{bmatrix} = \begin{bmatrix} x[0] & u[0] \\ x[1] & u[1] \\ \vdots & \vdots \\ x[m-1] & u[m-1] \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix}$$

$$\vec{s} = D\vec{p}$$

(b) What if there were now two distinct scalar inputs to a scalar system

$$x[i+1] = ax[i] + b_1 u_1[i] + b_2 u_2[i] + w[i] \tag{2}$$

and that we have measurements as before, but now also for both of the control inputs.

Set up a least-squares problem that you can solve to get an estimate of the unknown system parameters a, b_1, b_2.

Answer: Our new model is of the form

$$x[i+1] = ax[i] + b_1u_1[i] + b_2u_2[i] + w[i]$$

where $w(i)$ is our error term and we are interested in a and b.

We have $[1, m]$ measurements, and so our least squares formulation is:

$$\begin{bmatrix} x[1] \\ x[2] \\ \dots \\ x[m] \end{bmatrix} = \begin{bmatrix} x[0] & u_1[0] & u_2[0] \\ x[1] & u_1[1] & u_2[1] \\ \dots & \dots & \dots \\ x[m-1] & u_1[m-1] & u_2[m-1] \end{bmatrix} \begin{bmatrix} a \\ b_1 \\ b_2 \end{bmatrix}$$

$$\vec{s} = D\vec{p}$$

(c) What could go wrong in the previous case? For what kind of inputs would make least-squares fail to give you the parameters you want?

Answer: $D^\top D$ might not be invertible, which would cause this to fail. This happens when D has columns that are not linearly independent. For example, it could be because the inputs $\vec{u}_1$ and $\vec{u}_2$ are too similar, as if $\vec{u}_1 = \alpha\vec{u}_2$. We need these two inputs to be different and sufficiently varied so that least-squares does not fail.

(d) Returning to the scalar case with only one input, what could go wrong? When would you be unable to use least-squares to get the parameters you want?

Answer: As above, $D^\top D$ might not be invertible, which would cause this to fail. In this case, it might be because $\vec{x} = \alpha\vec{u}$. This can occur, for example, if $\vec{u} = 1$ always, $a = 0$ and $b = 1$, so that $\vec{x} = 1$ always as well. That is, the input $\vec{u}$ also needs to change in time in a way that is not just proportional to $x[0]$. Otherwise we can't untangle u and x.

(e) Now consider the two dimensional state case with a single input.

$$\vec{x}[i+1] = \begin{bmatrix} x_1[i+1] \\ x_2[i+1] \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \vec{x}[i] + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} u[i] + \vec{w}[i] \tag{3}$$

How can we treat this like two parallel problems to set this up using least-squares to get estimates for the unknown parameters $a_{11}, a_{12}, a_{21}, a_{22}, b_1, b_2$? What work/computation can we reuse across the two problems?

Answer: We can treat this as two parallel problems, with the first row and second row making up the two problems. Let r be the row index:

$$x_r[i+1] = \begin{bmatrix} a_r1 & a_r2 \end{bmatrix} \begin{bmatrix} x_1[i] \\ x_2[i] \end{bmatrix} + b_r u[i] + w_r[i]. \tag{4}$$

Combining terms to group the unknowns we can rewrite this as

$$x_r[i+1] = \begin{bmatrix} x_1[i] & x_2[i] & u[i] \end{bmatrix} \begin{bmatrix} a_{r1} \\ a_{r2} \\ b_r \end{bmatrix} + w_r[i]. \tag{5}$$

Thus each row produces a least squares problem of the form

$$\begin{bmatrix} x_r[1] \\ x_r[2] \\ \vdots \\ x_r[m] \end{bmatrix} = \begin{bmatrix} x_1[0] & x_2[0] & u[0] \\ x_1[1] & x_2[1] & u[1] \\ \vdots & \vdots & \vdots \\ x_1[m-1] & x_2[m-1] & u[m-1] \end{bmatrix} \begin{bmatrix} a_{r1} \\ a_{r2} \\ b_r \end{bmatrix}. \tag{6}$$

Notice that the constant matrix is shared by all of the subproblems, meaning that we only need to calculate $\left(A^\top A\right)^{-1} A^\top$ once. Thus we can stack all the subproblems horizontally:

$$\begin{bmatrix} x_1[1] & x_2[1] \\ x_1[2] & x_2[2] \\ \vdots & \vdots \\ x_1[m] & x_2[m] \end{bmatrix} = \begin{bmatrix} x_1[0] & x_2[0] & u[0] \\ x_1[1] & x_2[1] & u[1] \\ \vdots & \vdots & \vdots \\ x_1[m-1] & x_2[m-1] & u[m-1] \end{bmatrix} \begin{bmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \\ b_1 & b_2 \end{bmatrix} \tag{7}$$

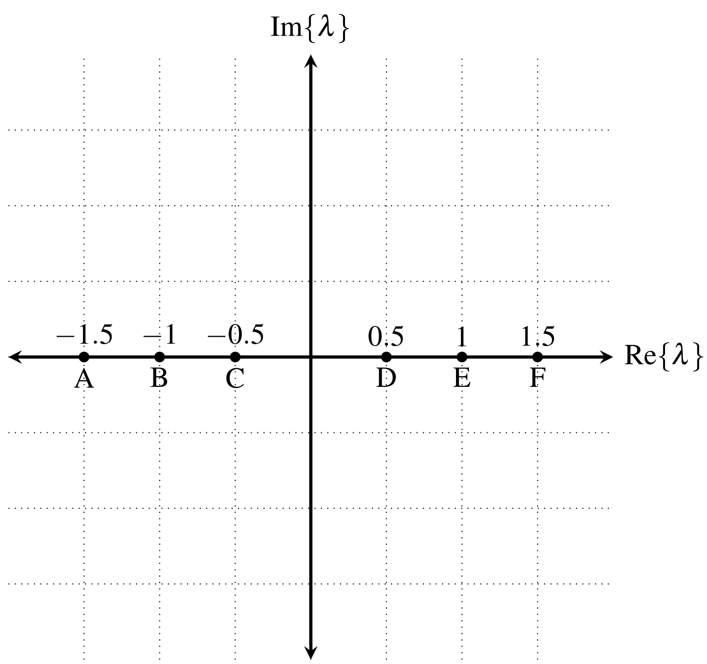
$$S = DP. \tag{8}$$

Finally, solving this as a single least squares problem gives us

$$P = \left(D^\top D\right)^{-1} D^\top S \tag{9}$$

2. Discrete time system responses

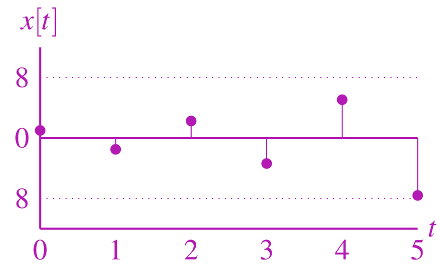
We have a system $x[k+1] = \lambda x[k]$. For each λ value plotted on the real-imaginary axis, sketch $x[k]$ with an initial condition of $x[0] = 1$. Determine if each system is stable.



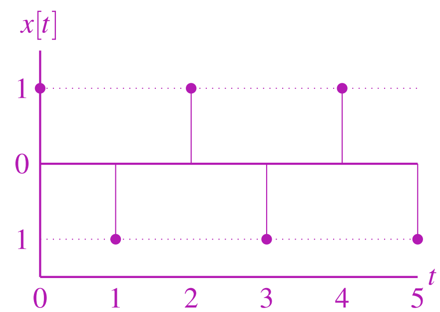
Answer: Solving our system for x[k] in terms of x[0] we get

$$x[k] = \lambda^k \cdot x[0]$$

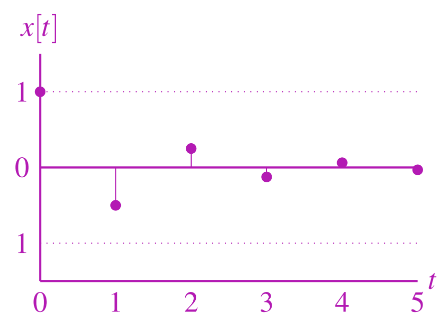
As you can see, for $|\lambda| > 1$ this expression will grow over time, for $|\lambda| < 1$ this expression will decay over time, and for $|\lambda| = 1$ this expression's growth will depend on the inputs.



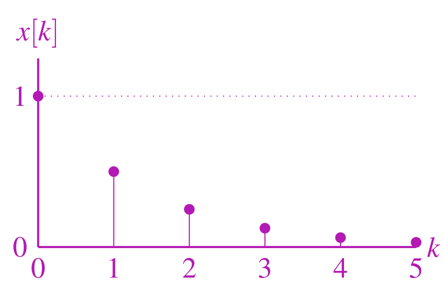
System A: Unstable
($\lambda = -1.5$)



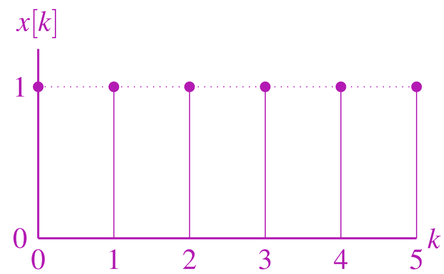
System B: Marginally Stable
($\lambda = -1.0$)



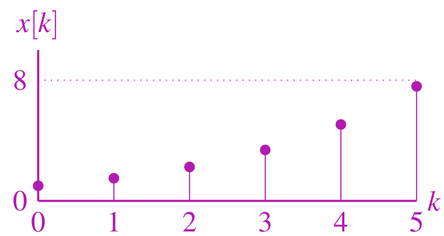
System C: Stable
($\lambda = -0.5$)



System D: Stable
($\lambda = 0.5$)



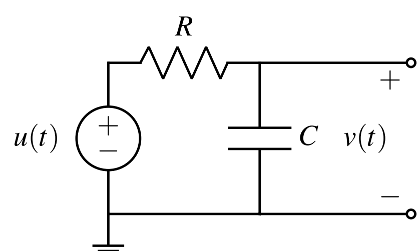
System E: Marginally Stable
($\lambda = 1.0$)



System F: Unstable
($\lambda = 1.5$)

3. Stability Examples and Counterexamples

(a) Consider the circuit below with $R = 1\Omega$, $C = 0.5\text{F}$, and $u(t) = \cos(t)$. Furthermore assume that $v(0) = 0$ (that the capacitor is initially discharged).



This circuit can be modeled by the differential equation

$$\frac{d}{dt}v(t) = -2v(t) + 2u(t) \tag{10}$$

Show that the differential equation is always stable. Consider what this means in the physical circuit.

Answer:

Let's try to make the differential equation $\frac{d}{dt}v(t) = -2v(t) + 2u(t)$ "blow up" (ie have $v(t)$ grow towards $+\infty$).

This means we want the right hand side to be positive:

$$-2v(t) + 2u(t) > 0 \qquad \Rightarrow \qquad u(t) > v(t)$$

But at some point, after $v(t)$ has grown to 1, we can no longer have $u > v$, as $u(t) = \cos(t)$ is bounded by $-1 < u(t) < 1$. Thus at some point, the right-hand-side will not be positive, so $v(t)$ cannot grow indefinitely. The system output is thus bounded in this case of a bounded input.

For the physical system, we can interpret this as: the voltage on the capacitor cannot ever exceed the voltage from the voltage source.

(b) Consider the discrete system

$$x[k+1] = 2x[k] + u[k] \tag{11}$$

with $x[0] = 0$.

Is the system stable or unstable? If unstable, find a bounded input sequence $u[k]$ that causes the system to "blow up". If unstable, is there still a (non-trivial) bounded input sequence that does not cause the system to "blow up"?

Answer: The system is unstable. This can be seen by considering the piecewise input

$$u[k] = 1, 0, 0, 0, 0, \ldots$$

k	0	1	2	3	$\ldots$
$x[k]$	0	1	2	4	$\ldots$
$u[]$	1	0	0	0	$\ldots$
$2x[k]+u[k]$	1	2	4	8	$\ldots$

There is still, however, a case for which a non-zero input results in a stable output:

$$u[k] = 1, -2, 0, 0, 0, 0, \ldots$$

k	0	1	2	3	$\ldots$
$x[k]$	0	1	0	0	$\ldots$
$u[k]$	1	-2	0	0	$\ldots$
$2x[k]+u[k]$	1	0	0	0	$\ldots$

In fact, there are an infinite number of input sequences that would result in stable outputs. But because we can find a single example of a bounded input sequence that leads to an unbounded output, the system in general is unstable.

Contributors:

- Anant Sahai.
- Regina Eckert.
- Kareem Ahmad.
- Kyle Tanghe.
- Sidney Buchbinder.